

REMARKS

Applicant thanks Examiner Vu for the in-person interview at the United States Patent and Trademark Office on January 31, 2007, with applicant's attorney, Mr. Uribe, and applicant's agent, Mr. Basrai. Applicant thanks Examiner Vu for his suggestions and explanation of the rejection of Claims 1-9, 11-17, and 19-25 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,625,804, issued to Ringseth et al. (hereinafter "Ringseth et al.") and the rejection of Claims 1, 10, 11, 18, 19, and 26 under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. 2003/0208743, to Chong et al. (hereinafter "Chong et al.") cited in the Final Office Action mailed January 12, 2007 (hereinafter "Office Action").

In response to the discussion of the above rejections and in conjunction with the Office Action remarks, applicant respectfully requests that the instant application be reexamined. Claims 1-26 were pending and are presented for examination herewith.

Objections to the Specification

The disclosure was objected to because of the following informalities: "the disclosure at different places mentions about an 'API call replay tool' and an 'API replay tool'; since there is no consistency as to when this above tool is just a replay tool as opposed to a call replay tool, absent any explicit definition to each concept, there is no clear way to construe this tool as being one or another." In response, applicant has made several clarifying amendments to the specification to explicitly define the concept of the API replay tool and to be consistent with the language. Applicant also noticed other typographical errors in the specification and has amended the specification accordingly.

Objections to the Claims

Claim 1 was objected to due to a typographical error. In response, applicant has made appropriate corrections to obviate the objection. Accordingly, applicant submits that the objection has been rendered moot and requests that the objection of Claim 1 be withdrawn.

Claims 1, 11, and 19 were objected to as containing inconsistencies. In response, applicant has made appropriate amendments to Claims 1, 11, and 19 and to several of the claims that depend from Claims 1, 11, and 19. Accordingly, applicant submits that the objection has been rendered moot and requests that the objection to Claims 1, 11, and 19 be withdrawn.

Claim Rejections

As noted above, the Office Action rejected Claims 1-9, 11-17, and 19-25 under 35 U.S.C. § 102(e) as being anticipated by Ringseth et al. The Office Action further rejected Claims 1, 10, 11, 18, 19, and 26 under 35 U.S.C. § 102(e) as being anticipated by Chong et al. Applicants respectfully disagree because Ringseth et al. and Chong et al. do not teach, suggest, or describe all claim limitations, which are discussed in detail hereinafter.

Prior to discussing in detail why applicants believe that all of the claims in the present application are allowable in view of the cited and applied reference, a brief description of the disclosed subject matter and a brief description of the teachings of the cited and applied references are provided. The following discussions of the disclosed subject matter and the cited and applied reference are not provided to define the scope or interpretation of any of the claims of this application. Instead, these discussions are provided to help the United States Patent and Trademark Office better appreciate important claim distinctions discussed thereafter.

Disclosed Subject Matter

Generating and executing a sequence of API calls for a variety of purposes, including testing the APIs on a new platform or diagnosing an error through the use of a software tool, is disclosed. The software tool facilitates submitting API calls corresponding to previously logged API calls to various components of a computing device. This is generally accomplished by translating memory references specified by the logged API calls into memory references within the software tool's memory space. More particularly, the software tool includes symbol tables for mapping memory references within logged API calls with corresponding memory references within returns of logged API calls submitted by the software tool. Such mapping occurs, for

example, from a recorded address to a replay address space allocated to a thread with which the API call is associated. The mapped memory references are used for submitting subsequent API calls corresponding to the logged API calls from within the memory space of the software tool to the various components of the computing device. After mapping the memory references into the software tool's memory space, an API call builder utilizes the mapped memory references stored within a symbol table to create an API call code sequence for submitting the subsequent API calls corresponding to the logged API calls to the various components of the computing device.

Summary of U.S. Patent No. 6,625,804 (Ringseth et al.)

Ringseth et al. purportedly discloses a unified event programming model that standardizes event programming for disparate eventing protocols. The unified event programming model simplifies programming events for different object types by abstracting away protocol-specific details. A protocol-independent compiler construct allows a programmer to specify events for an event source. Other protocol-independent compiler constructs allow a programmer to specify how to hook and unhook an event receiver from events. Based upon protocol-independent compiler constructs and an eventing protocol type value, a compiler generates an event source or event receiver implementation that is specific to an eventing protocol. Ringseth et al. does not disclose mapping memory references within logged API calls with corresponding memory references within returns of logged API calls submitted by the software tool, where the mapped memory references are used for submitting subsequent API calls corresponding to the logged API calls from within the memory space of the software tool to the various components of the computing device, and an API call builder for creating an API call code sequence for submitting subsequent API calls corresponding to the logged API calls to the various components of the computing device according to the mapped memory references.

Summary of U.S. Patent Application Publication No. 2003/0208743 (Chong et al.)

Chong et al. purportedly discloses a workflow code generator for generating executable code for multi-channel and/or multi-modal applications. The code generator may include a

parser for reading application input files and creating internal representations of declarative statements within the input files. The code generator may further include a model analyzer, which processes the internal model to detect errors, perform optimization, and prepare for outputting the result. The code generator uses a symbol or mapping table for storing references to resources that have been used by the input application. The code generator assigns code fragments to object patterns, resolves data object references by referring to a mapping table, and traverses the objects and emits code assigned to the objects. Chong et al. does not disclose mapping memory references within logged API calls with corresponding memory references within returns of logged API calls submitted by the software tool, where the mapped memory references are used for submitting subsequent API calls corresponding to the logged API calls from within the memory space of the software tool to the various components of the computing device, and an API call builder for creating an API call code sequence for submitting subsequent API calls corresponding to the logged API calls to the various components of the computing device according to the mapped memory references.

Rejection of Claims 1-9, 11-17, and 19-25 Under 35 U.S.C. § 102(e)

As indicated above, Claims 1-9, 11-17, and 19-25 were rejected under 35 U.S.C. § 102(e) as being anticipated by Ringseth et al.

Claims 1-9

As amended, independent Claim 1 reads as follows:

1. A computing device containing a software tool for submitting API calls corresponding to logged API calls to various components of the computing device, the software tool comprising:

symbol tables for mapping memory references within logged API calls with corresponding memory references within returns of logged API calls submitted by the software tool, wherein the mapped memory references are used for submitting subsequent API calls corresponding to the logged API calls from within the memory space of the software tool to the various components of the computing device; and

an API call builder for creating an API call code sequence for submitting subsequent API calls corresponding to logged API calls to the

various components of the computing device according to the mapped memory references within a symbol table.

Applicant submits that Ringseth et al. does not teach or suggest "mapping memory references within logged API calls with corresponding memory references within returns of logged API calls submitted by the software tool, wherein the mapped memory references are used for submitting subsequent API calls corresponding to the logged API calls from within the memory space of the software tool to the various components of the computing device" and accordingly does not teach "creating an API call code sequence for submitting subsequent API calls corresponding to logged API calls to the various components of the computing device according to the mapped memory references" as recited in Claim 1.

In contrast, Ringseth et al. teaches a symbol table that is used to list symbol names and type information used in a source file with UEPM constructs along with related characteristics. (Col. 11, lines 53-56.) However, Ringseth et al. fails to teach or suggest mapping memory references within logged API calls with corresponding memory references within returns of logged API calls submitted by the software tool, such that mapped memory references are used for submitting subsequent API calls corresponding to the logged API calls from within the memory space of the software tool to the various components of the computing device as recited in Claim 1. The symbol table of Ringseth is not utilized for the translation of memory references of logged API calls.

Since Ringseth et al. does not disclose mapping memory references within logged API calls with corresponding memory references within returns of logged API calls submitted by the software tool, Ringseth et al. cannot further disclose that these mapped memory references are specified for creating an API call code sequence for submitting subsequent API calls corresponding to logged API calls to the various components of the computing device. Instead, Ringseth et al. is directed to teaching an interface for the symbol table. The interface, however, does not teach a return of logged API calls.

Since Ringseth et al. does not teach, suggest, or describe all of the limitations of Claim 1, applicant requests withdrawal of the rejection of Claim 1 under 35 U.S.C. § 102(e) based on Ringseth et al. and allowance of the claim. Claims 2-9 depend directly or indirectly from independent Claim 1 and include all of the limitations of the base claim. Accordingly, Claims 2-9 are submitted to be allowable for at least the same reasons that Claim 1 is allowable.

Claims 11-17

As amended, independent Claim 11 reads as follows:

11. A method for invoking API calls based on logged API calls, the method comprising:
 - mapping memory references within logged API calls with corresponding memory references within returns of logged API calls invoked by a software tool, wherein the mapped memory references are used for invoking subsequent API calls based on the logged API calls from within the memory space of the software tool; and
 - creating an API call code sequence for invoking the subsequent API calls based on the logged API calls, wherein the mapped memory references are specified for invoking the subsequent API calls.

Similar to Claim 1, Ringseth et al. does not disclose "mapping memory references within logged API calls with corresponding memory references within returns of logged API calls invoked by a software tool, wherein the mapped memory references are used for invoking subsequent API calls based on the logged API calls from within the memory space of the software tool" as recited in Claim 11. Accordingly, Ringseth et al. does not disclose "creating an API call code sequence for invoking the subsequent API calls based on the logged API calls, wherein the mapped memory references are specified for invoking the subsequent API calls" as also recited in Claim 11.

Since Ringseth et al. does not teach, suggest, or describe all of the limitations of Claim 11, applicant requests withdrawal of the rejection of Claim 11 under 35 U.S.C. § 102(e) based on Ringseth et al. and allowance of the claim. Claims 12-17 depend directly or indirectly from independent Claim 11 and include all of the limitations of the base claim. Accordingly,

Claims 12-17 are submitted to be allowable for at least the same reasons that Claim 11 is allowable.

Claims 19-25

Independent Claim 19, as amended, reads as follows:

19. A computer-readable storage medium including computer-executable instructions for invoking API calls based on logged API calls to various components of a computing device, the computer-executable instructions performing the steps of:

mapping memory references within logged API calls with corresponding memory references within returns of logged API calls invoked by a software tool, wherein the mapped memory references are used for invoking subsequent API calls based on the logged API calls from within the memory space of the software tool; and

creating an API call code sequence for invoking the subsequent API calls based on the logged API calls, wherein the mapped memory references are specified for invoking the subsequent API calls.

Similar to Claims 1 and 11, Ringseth et al. does not disclose "mapping memory references within logged API calls with corresponding memory references within returns of logged API calls invoked by a software tool, wherein the mapped memory references are used for invoking subsequent API calls based on the logged API calls from within the memory space of the software tool" as recited in Claim 19. Accordingly, Ringseth et al. does not disclose "creating an API call code sequence for invoking the subsequent API calls based on the logged API calls, wherein the mapped memory references are specified for invoking the subsequent API calls" as also recited in Claim 19.

Since Ringseth et al. does not teach, suggest, or describe all of the limitations of Claim 19, applicant requests withdrawal of the rejection of Claim 19 under 35 U.S.C. § 102(e) based on Ringseth et al. and allowance of the claim. Claims 20-25 depend directly or indirectly from independent Claim 19 and include all of the limitations of the base claim. Accordingly, Claims 20-25 are submitted to be allowable for at least the same reasons that Claim 19 is allowable.

Rejection of Claims 1, 10, 11, 18, 19, and 26 Under 35 U.S.C. § 102(e)

As indicated above, Claims 1, 10, 11, 18, 19, and 26 were rejected under 35 U.S.C. § 35 U.S.C. § 102(e) as being anticipated by Chong et al.

Claims 1 and 10

Applicant submits that Chong et al. does not teach each and every limitation of independent Claim 1. Specifically, Chong et al. does not teach "mapping memory references within logged API calls with corresponding memory references within returns of logged API calls submitted by the software tool, wherein the mapped memory references are used for submitting subsequent API calls corresponding to the logged API calls from within the memory space of the software tool to the various components of the computing device" and accordingly does not teach "creating an API call code sequence for submitting subsequent API calls corresponding to logged API calls to the various components of the computing device according to the mapped memory references" as recited in Claim 1.

In contrast to Claim 1, Chong et al. teaches a symbol table 118 of code generator 110 for storing references to resources that have been used by the input application. (Paragraph 0109, page 8.) The code generator reads the input file and converts the declarative statements into an intermediate form that is more easily understood by using machine code. The code generator generates code from XML files, which is an in-memory image of objects that comprise a workflow. The objects include workflow variables, states, transitions, etc. (Paragraph 0113, page 8.) Nowhere, however, does Chong et al. disclose mapping memory references within logged API calls with corresponding memory references within returns of logged API calls submitted by the software tool, where the mapped memory references are used for submitting subsequent API calls corresponding to the logged API calls from within the memory space of the software tool to the various components of the computing device.

Since Chong et al. does not disclose mapping memory references within logged API calls with corresponding memory references within returns of logged API calls submitted by the

software tool, Chong et al. cannot further disclose that these memory references are specified for creating an API call code sequence for submitting subsequent API calls corresponding to logged API calls to the various components of the computing device.

Since Chong et al. does not teach, suggest, or describe all of the limitations of Claim 1, applicant requests withdrawal of the rejection of Claim 1 under 35 U.S.C. § 102(e) based on Chong et al. and allowance of the claim. Claim 10 depends directly from independent Claim 1 and includes all of the limitations of the base claim. Accordingly, Claim 10 is submitted to be allowable for at least the same reason that Claim 1 is allowable.

Claims 11 and 18

Similar to Claim 1, Chong et al. does not disclose "mapping memory references within logged API calls with corresponding memory references within returns of logged API calls invoked by a software tool, wherein the mapped memory references are used for invoking subsequent API calls based on the logged API calls from within the memory space of the software tool" recited in Claim 11. Accordingly, Chong et al. does not disclose "creating an API call code sequence for invoking the subsequent API calls based on the logged API calls, wherein the mapped memory references are specified for invoking the subsequent API calls" as also recited in Claim 11.

Since Chong et al. does not teach, suggest, or describe the foregoing limitation of Claim 11, applicant requests withdrawal of the rejection of Claim 11 under 35 U.S.C. § 102(e) based on Chong et al. and allowance of the claim. Claim 18 depends directly from independent Claim 11 and includes all of the limitations of the base claim. Accordingly, Claim 18 is submitted to be allowable for at least the same reasons that Claim 11 is allowable.

Claims 19 and 26

Similar to Claims 1 and 11, Chong et al. does not disclose "mapping memory references within logged API calls with corresponding memory references within returns of logged API calls invoked by a software tool, wherein the mapped memory references are used for invoking

subsequent API calls based on the logged API calls from within the memory space of the software tool" recited in Claim 19. Accordingly, Chong et al. does not disclose "creating an API call code sequence for invoking the subsequent API calls based on the logged API calls, wherein the mapped memory references are specified for invoking the subsequent API calls" as also recited in Claim 19.

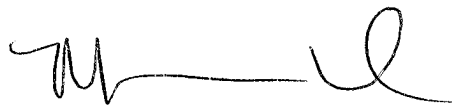
Since Chong et al. does not teach, suggest, or describe the foregoing limitation of Claim 19, applicant requests withdrawal of the rejection of Claim 19 under 35 U.S.C. § 102(e) based on Chong et al. and allowance of the claim. Claim 26 depends directly from independent Claim 19 and includes all of the limitations of the base claim. Accordingly, Claim 26 is submitted to be allowable for at least the same reason that Claim 19 is allowable.

CONCLUSION

In view of the foregoing remarks, applicant respectfully submits that the above-identified application is in condition for allowance. Reconsideration and reexamination of the application, and allowance of the claims (Claims 1-26) at an early date are solicited. If the Examiner has any questions or comments concerning this matter, the Examiner is invited to contact the undersigned at the number provided below.

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS^{PLLC}



Mauricio A. Uribe
Registration No. 46,206
Direct Dial No. 206.695.1728

MAU/ITXB:tmn

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue, Suite 2800
Seattle, Washington 98101
206.682.8100